|  |  |
| --- | --- |
| Wydział Informatyki  Przedmiot: Systemy Wbudowane | Data  28.02.2019 |
| Ćwiczenie nr. 1  Temat: Zapoznanie się z CAD Quartus i płytą prototypową DE1-SoC  Grupa PS 6  Adam Polejczuk  Damian Rosiński | Prowadzący:  prof. dr hab. inż.  Valery Salauyou  Ocena:  …………… |

**Cel zajęć:**

Zapoznanie się z laboratorium, oprogramowaniem Quartus II 17.1 Prime, jęzukiem Verilog HDL służącym do projektowania oraz symulacji układów cyfrowych, oraz płytką programowalną DE1\_SoC na której zrealizowaliśmy poszczególne zadania.

**Informacje wstępne:**

Tabela prawdy dla realizowanych sumatorów jednobitowych.

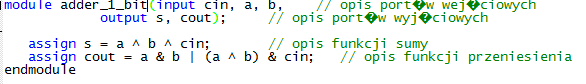
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Wejścia | | | Wyjścia | |
| CIN | A | B | COUT | S |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 |

*Rys.1.0 Tablica prawdy sumatora 1-bit.*

**Zadanie 1:**

Zaimplementowanie projektu **jednobitowego sumatora** przy pomocy operatorów **assign**.

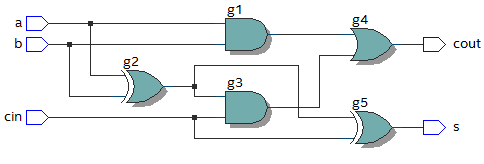
Implementacja programu realizującego sumator:



*Rys.1.1 Realizacja sumatora jednobitowego.*

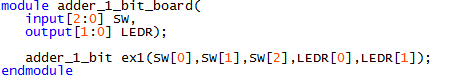
Stworzony powyższy modół adder\_1\_bit realizuje sumator jednobitowy przy pomocy operatorów przypisania **assign**.

Schemat Stworzonego sumatora jednobitowego w języku Verilog:



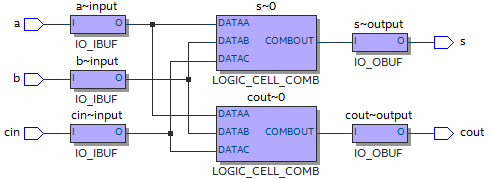
*Rys.1.2 Realizacja sumatora jednobitowego na bramkach logicznych.*

Implementacja programu realizującego sumator jednobitowy na płytce DE1\_Soc:



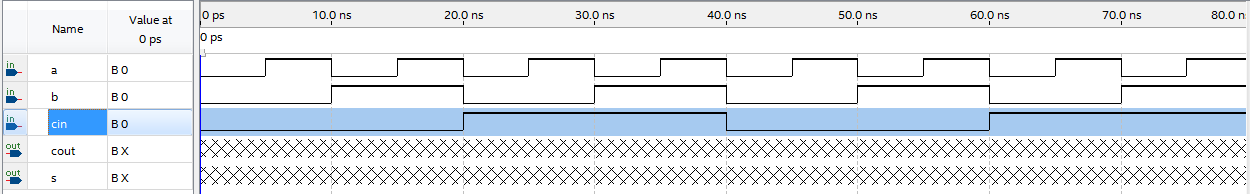
*Rys.1.3 Implementacja sumatora jednobitowego dla układu cyfrowego DE1\_SoC.*

Schemat wynikowy sumatora jednobitowego dla układu DE1\_SoC:

****

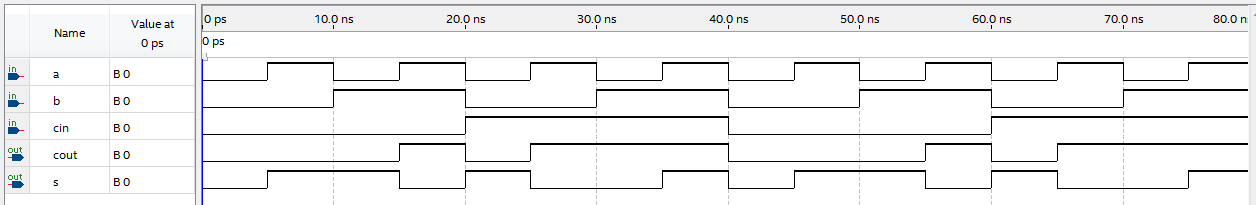
*Rys.1.4 Realizacja sumatora jednobitowego dla układu cyfrowego DE1\_SoC.*

Początkowe ustawienie zegarów poszczególnych wejść a, b, cin na wartości 10ns, 20ns, 40ns.

****

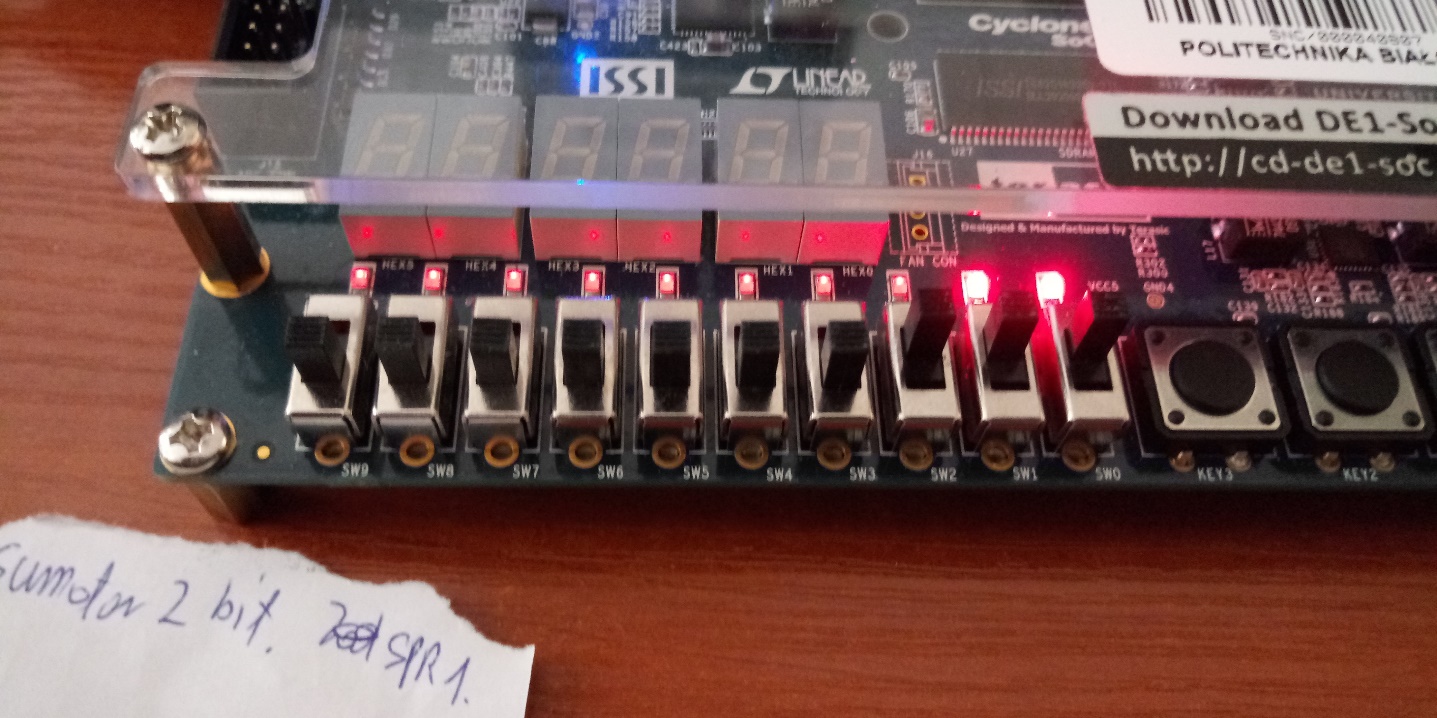
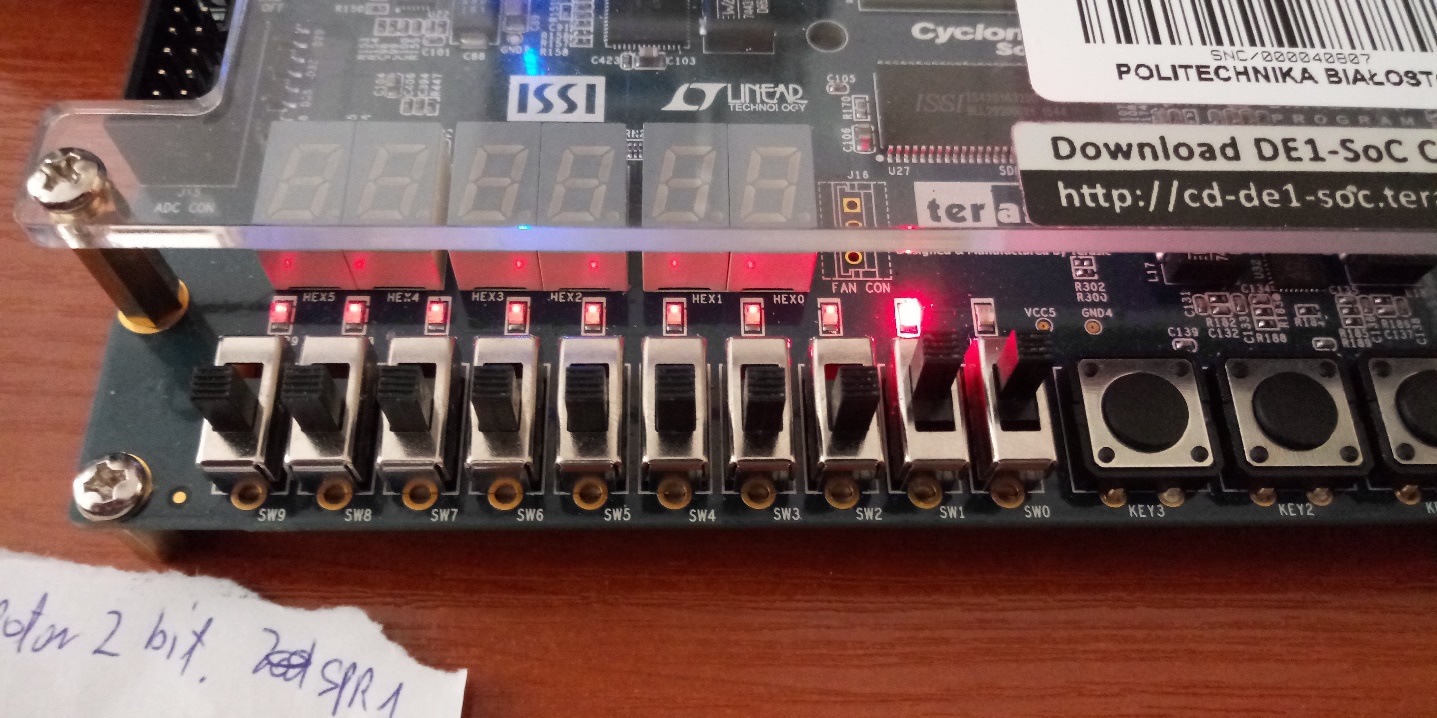
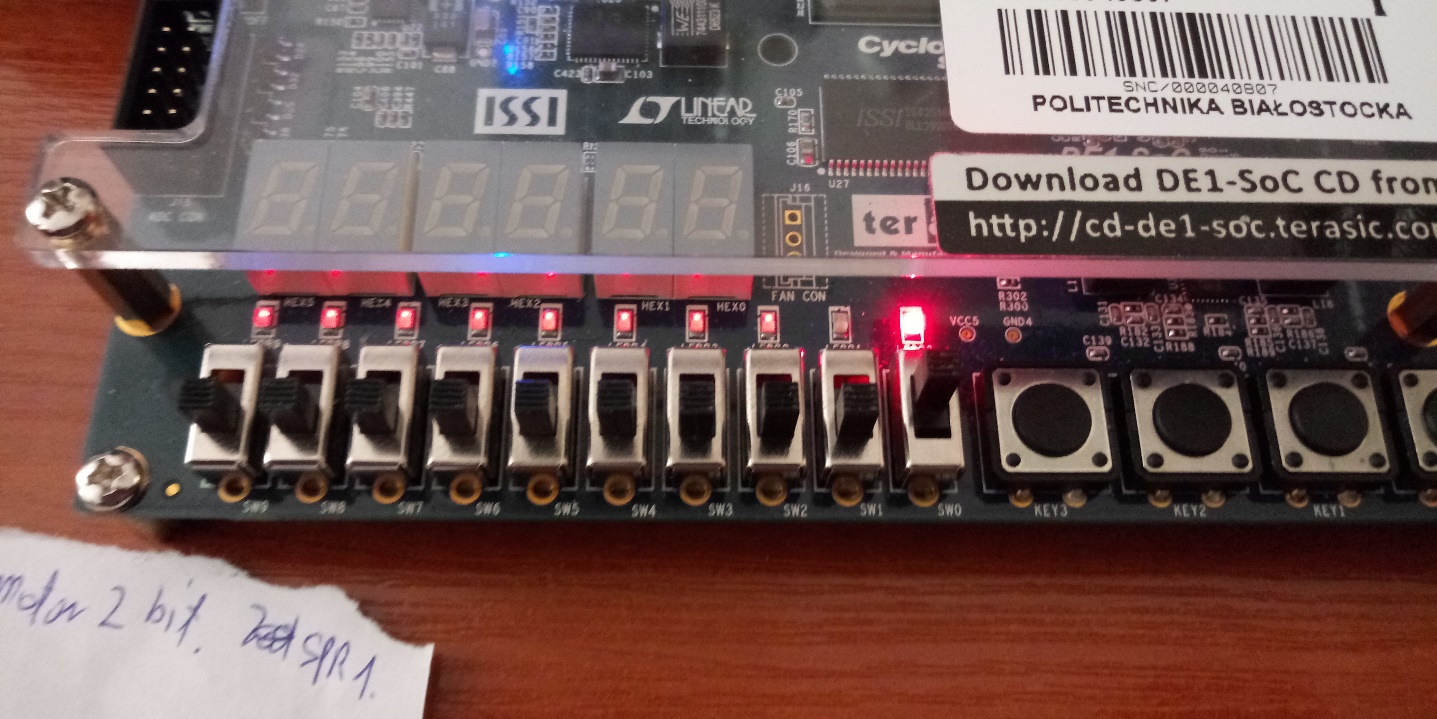
*Rys.1.5 Ustawienie wartości zegarów.*

Po przeprowadzeniu symulacji na układzie, otrzymaliśmy następujący wynik:

****

*Rys.1.5 Wynik przeprowadzonej symulacji dla układu DE1\_SoC.*

Następnie przetestowaliśmy działanie naszego projektu na układzie DE1\_SoC:

****

**Rozwiązanie:**

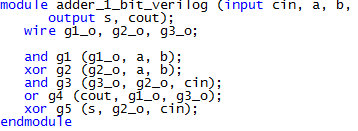
Z początku stworzyliśmy algorytm realizujący sumator jednobitowy po czym dołączyliśmy niezbędne pliki do działania programu i skompilowaliśmy projekt.

Kolejnym krokiem było stworzenie implementacji dla układu DE1\_SoC, aby zsynchronizować wcześniej stworzony program z odpowiednimi przełącznikami oraz diodami na płytce. Następnie skompilowaliśmy projekt i sprawdziliśmy jego działanie zgodnie z Rys.1.0.

**Zadanie 2:**

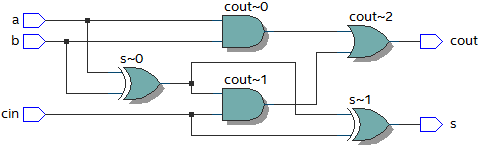
Zaimplementowanie projektu **jednobitowego sumatora** za pomocą **prymitywów** języka Verilog.

Implementacja programu realizującego sumator:



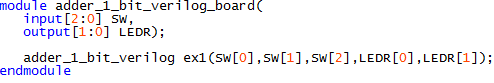
*Rys.2.1 Realizacja sumatora jednobitowego.*

Schemat Stworzonego sumatora jednobitowego w języku Verilog:



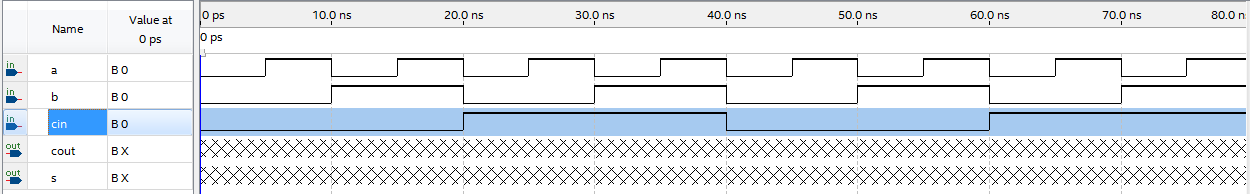
*Rys.2.2 Realizacja sumatora jednobitowego na bramkach logicznych.*

Implementacja programu realizującego sumator jednobitowy na płytce DE1\_Soc:



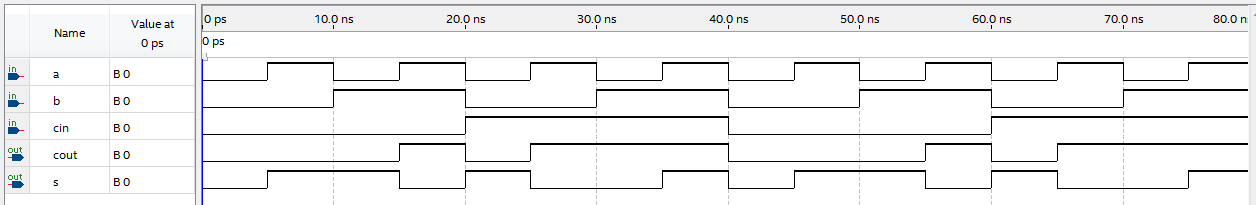
*Rys.2.3 Implementacja sumatora jednobitowego dla układu cyfrowego DE1\_SoC.*

Początkowe ustawienie zegarów poszczególnych wejść a, b, cin na wartości 10ns, 20ns, 40ns.

****

*Rys.2.4 Ustawienie wartości zegarów.*

Po przeprowadzeniu symulacji na układzie, otrzymaliśmy następujący wynik:

****

*Rys.2.5 Wynik przeprowadzonej symulacji dla układu DE1\_SoC.*

**Rozwiązanie:**

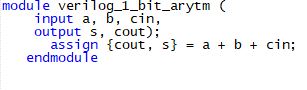
Podobnie ja w zadaniu 1 najpierw stworzyliśmy algorytm realizujący sumator jednobitowy po czym dołączyliśmy niezbędne pliki do działania programu i skompilowaliśmy projekt.

Kolejnym krokiem było stworzenie implementacji dla układu DE1\_SoC, aby zsynchronizować wcześniej stworzony program z odpowiednimi przełącznikami oraz diodami na płytce. Spodziewaliśmy się otrzymania takiego samego wyniku jak w zadaniu poprzednim. Następnie skompilowaliśmy projekt i sprawdziliśmy jego działanie zgodnie z Rys.1.0 i okazało się poprawnie.

**Zadanie 3:**

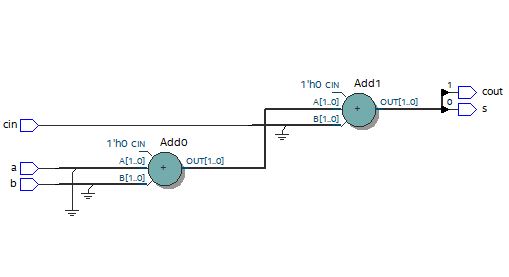
Zaimplementuj projekt jednobitowego sumatora za pomocą operacji arytmetycznej "+".

Implementacja programu realizującego sumator:



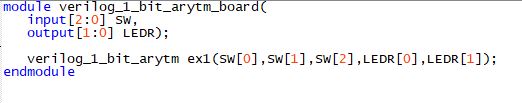
*Rys.2.1 Realizacja sumatora jednobitowego.*

Schemat Stworzonego sumatora jednobitowego w języku Verilog:



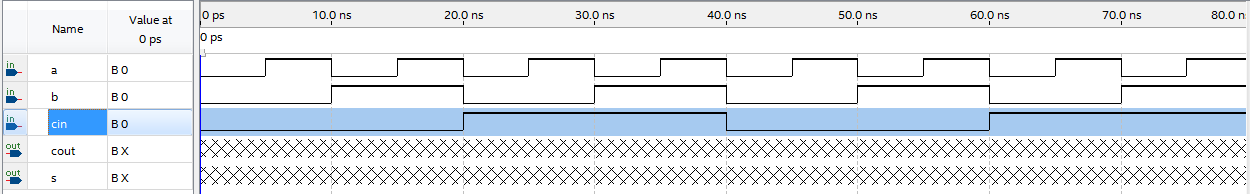
*Rys.2.2 Realizacja sumatora jednobitowego za pomocą operacji „+”.*

Implementacja programu realizującego sumator jednobitowy na płytce DE1\_Soc:



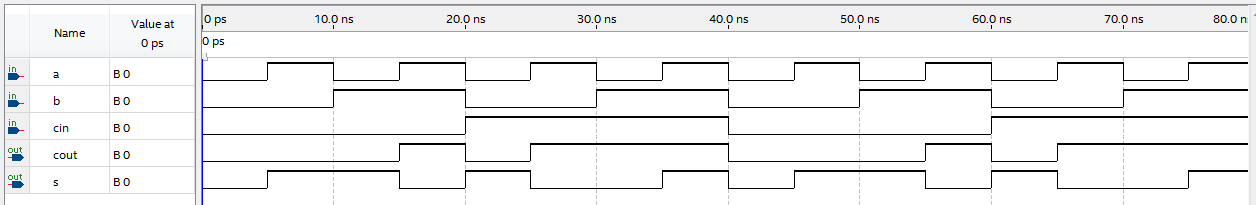
*Rys.2.3 Implementacja sumatora jednobitowego dla układu cyfrowego DE1\_SoC.*

Początkowe ustawienie zegarów poszczególnych wejść a, b, cin na wartości 10ns, 20ns, 40ns.

****

*Rys.2.4 Ustawienie wartości zegarów.*

Po przeprowadzeniu symulacji na układzie, otrzymaliśmy następujący wynik:

****

*Rys.2.5 Wynik przeprowadzonej symulacji dla układu DE1\_SoC.*

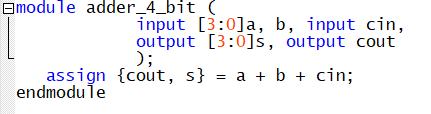
**Zadanie 4:**

Zaimplementuj 4-bitowy iteracyjny sumator z przeniesieniami (ripple carry) z następującymi

pinami:

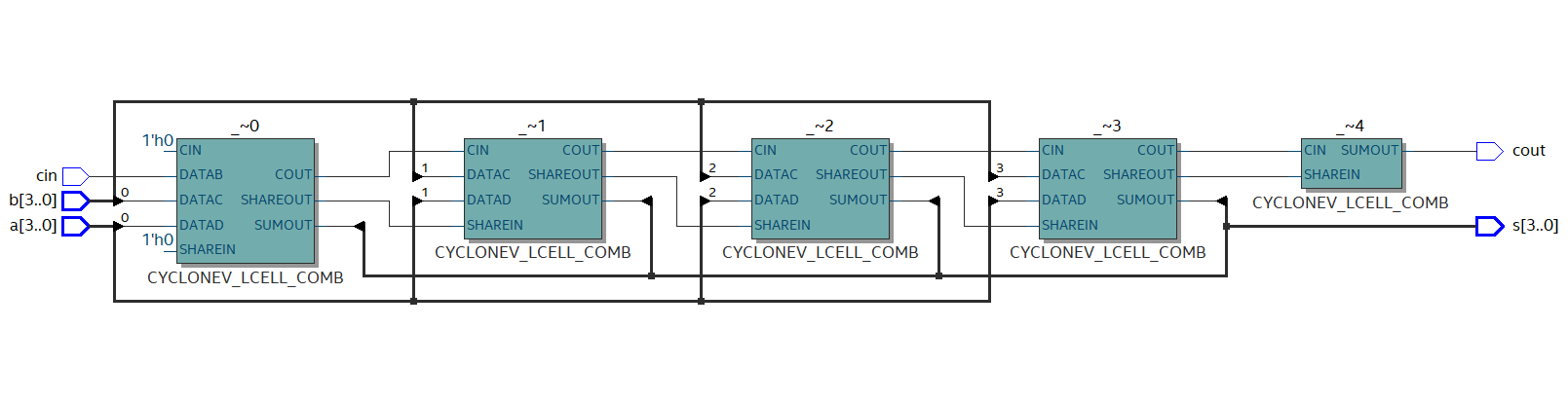
|  |  |
| --- | --- |
| Wyprowadzenia | Piny |
| A[3:0] | SW[3:0] |
| B[3:0] | SW[7:4] |
| cin | SW[9] |
| S[3:0] | LEDR[3:0] |
| cout | LEDR[9] |

Implementacja programu realizującego 4-bitowy sumator iteracyjny z przeniesieniami:



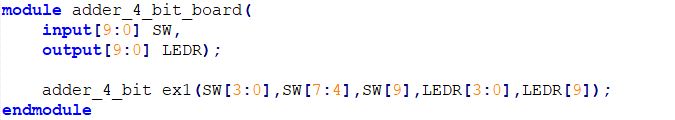
*Rys.4.1 Realizacja sumatora czterobitowego.*

Schemat Stworzonego sumatora czterobitowego w języku Verilog:



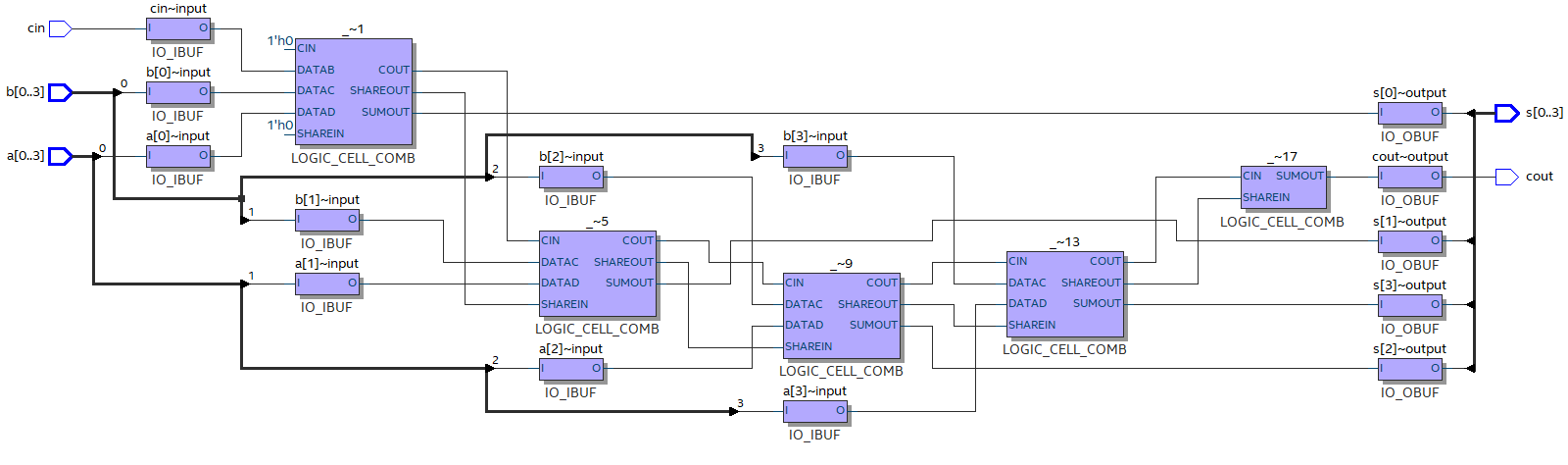
*Rys.4.2 Realizacja sumatora czterobitowego*

Implementacja programu realizującego sumator czterobitowy na płytce DE1\_Soc:



*Rys.4.3 Implementacja sumatora czterobitowego dla układu cyfrowego DE1\_SoC.*

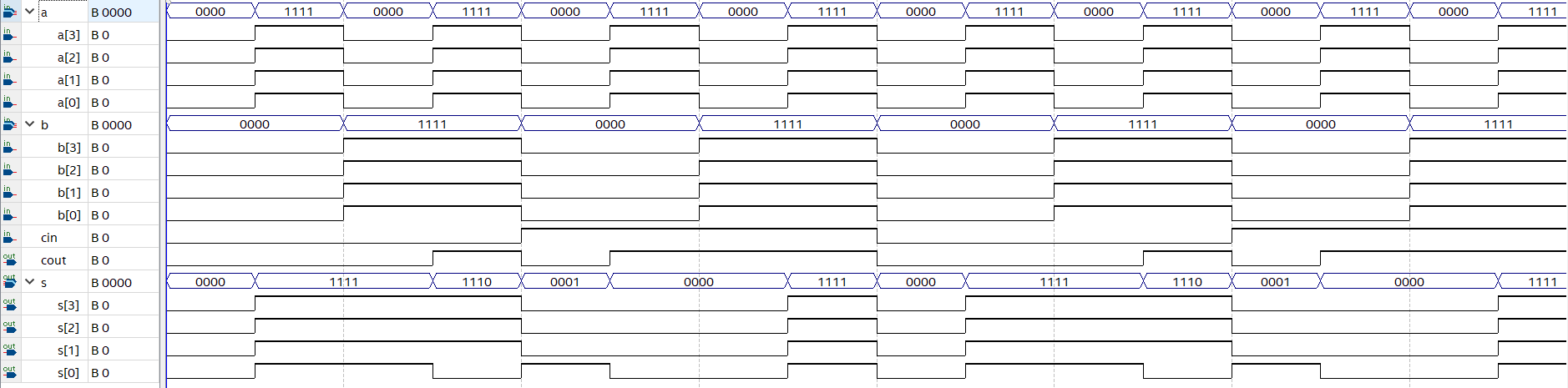
Schemat wynikowy sumatora czterobitowego dla układu DE1\_SoC:

****

*Rys.4.4 Realizacja sumatora czterobitowego dla układu cyfrowego DE1\_SoC.*

Początkowe ustawienie zegarów poszczególnych wejść a, b, cin na wartości 10ns, 20ns, 40ns.

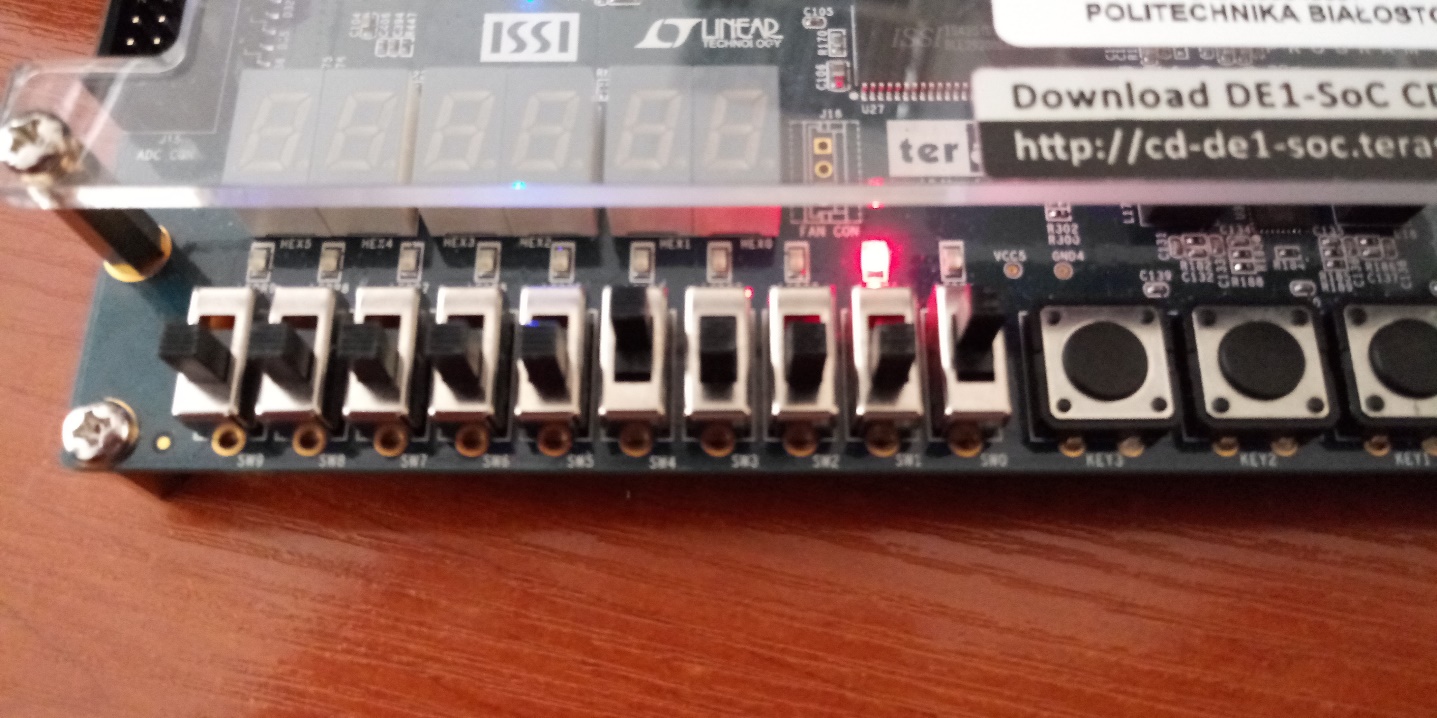
Po przeprowadzeniu symulacji na układzie, otrzymaliśmy następujący wynik:

****

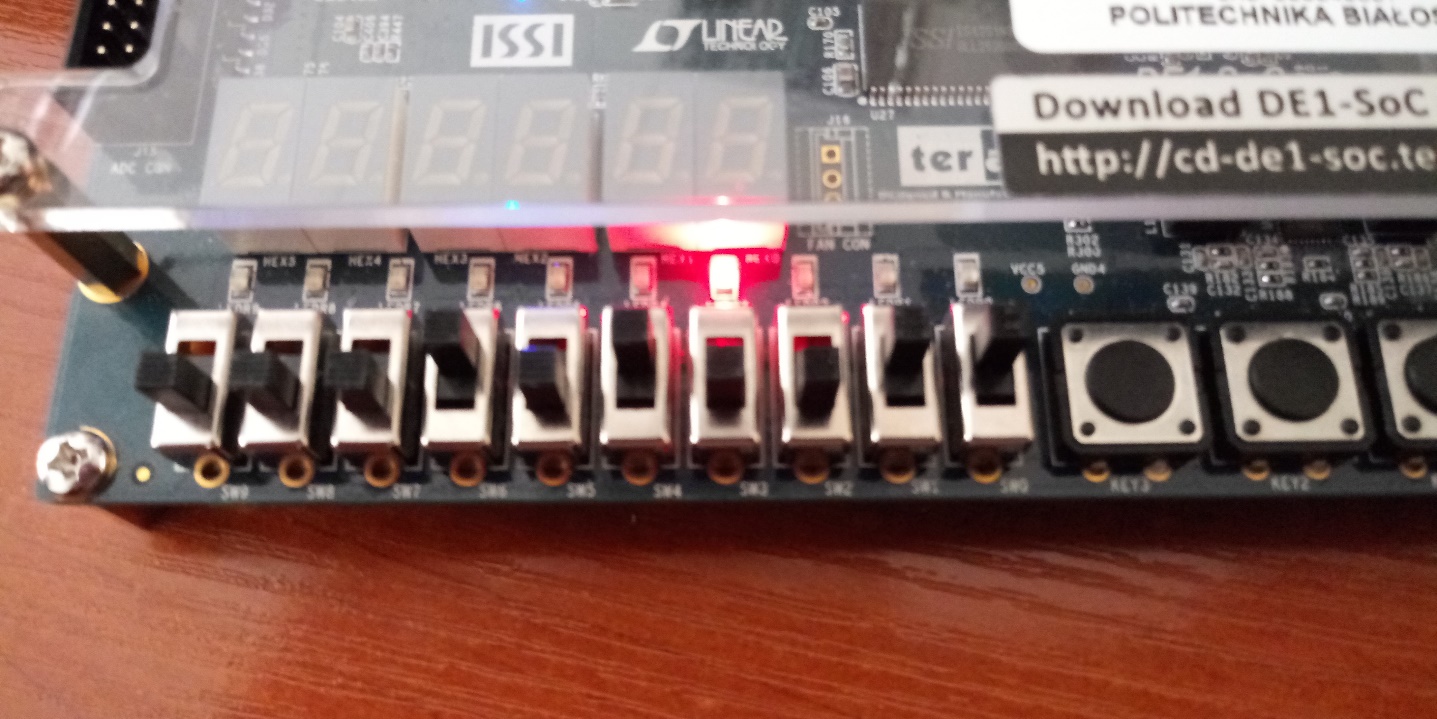
*Rys.4.5 Wynik przeprowadzonej symulacji dla układu DE1\_SoC.*

Następnie przetestowaliśmy działanie naszego projektu na układzie DE1\_SoC:

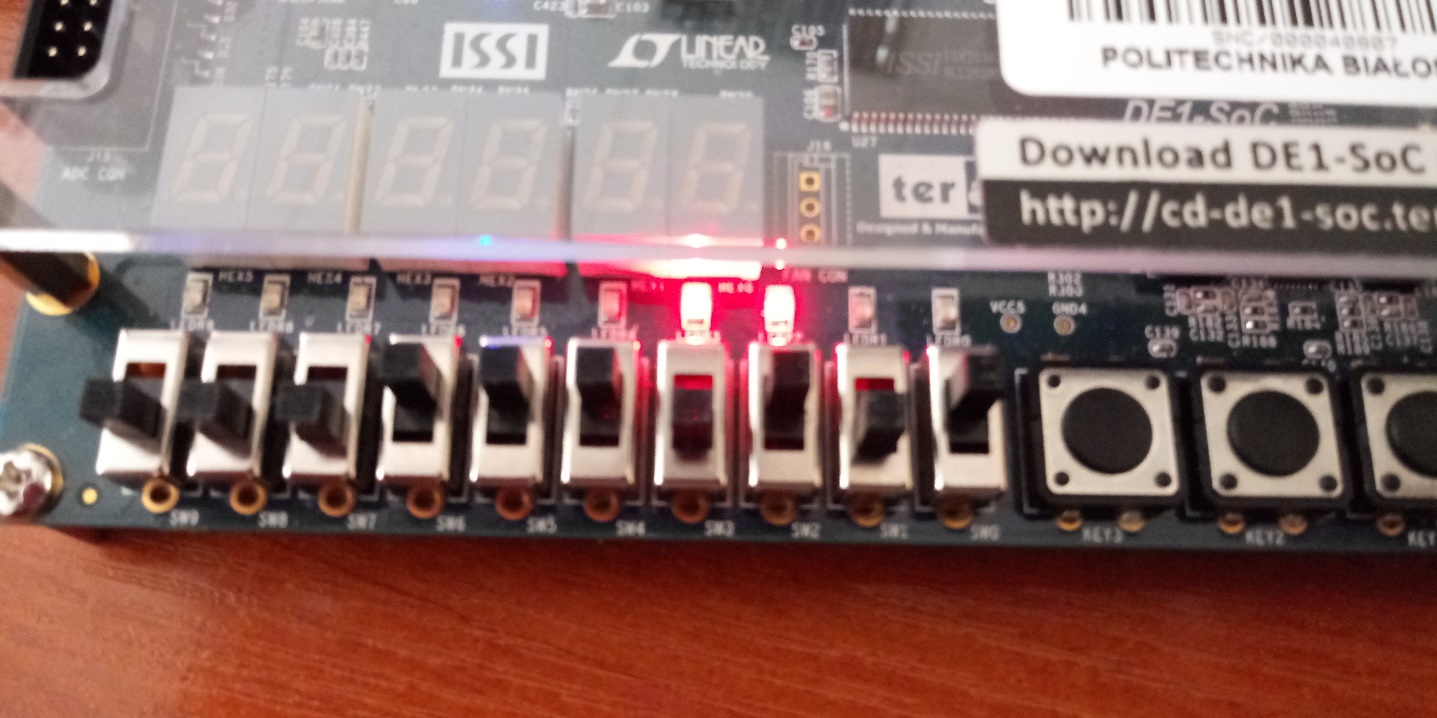
1+1=2



3+5=8



5+7=12



**Wnioski:**

* Na podstawie przeprowadzonej analizy układu cyfrowego DE1\_SoC**,** możemy stwierdzić, że za jego pomocą można zbudować układ realizujący program sumatora jednobitowego w próżnych jego implementacjach.
* Po zrealizowaniu zadania ze stworzeniem sumatora czterobitowego, zauważyliśmy, że jest maksymalny możliwy do zrealizowania sumator bitowy ze względu na ograniczenia fizyczne jakimi jest liczba przełączników (SW) oraz diod (LEDR).
* Podczas realizacji tworzenia sumatorów zauważyliśmy, że stan wyjściowy zależy od stanu wejściowego, czyli tak jak dla wyżej założonej tabeli prawdy. Do stworzenia sumatora N-bitowego wystarczy powielanie implementacji sumatora jednobitowego z drobną modyfikacją.